

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A process for transferring pen data between unmanaged and managed code on a computing device, the unmanaged code being code native to and executed directly by a processor of the computing device, the managed code being executed in a common language run-time environment of a framework operating on the computing device, the common language run-time environment of the framework executing the managed code independent of a type of the processor of the computing device, the process comprising the steps of:

receiving pen data in a pen input component on the computing device written in unmanaged code, the pen data being generated by a digitizer of the computing device upon movement of a stylus with respect to a surface of the digitizer, the pen data including at least one location on the digitizer of the stylus;

transferring information related to said pen data to a mutual exclusion shared memory on the computing device designated to be non-simultaneously shared between unmanaged code and managed code;

transferring, by the pen input component written in unmanaged code, a pointer that points to said information in said shared memory to a stylus input subsystem of an application on the computing device, the stylus input subsystem being separate from the pen input component and being written ~~at least in part~~ in managed code; and

retrieving, by the stylus input subsystem application written ~~at least in part~~ in managed code, said information from said shared memory by way of the transferred pointer.

2. (Original) The process according to claim 1, further comprising the steps of:
transferring additional information from said at least in part managed application to said shared memory;

transferring a pointer that points to said additional information to said component;
retrieving said additional information from said shared memory.

3. (Original) The process according to claim 1, further comprising the step of:
using a P-invoke style interface crossing between unmanaged code and managed code.
4. (Original) The process according to claim 1, further comprising the step of:
exchanging information through a COM interface.
5. (Original) The process according to claim 1, said component being a pen services
component.
6. (Original) The process according to claim 1, said application including a pen
input managed client.
7. (Original) The process according to claim 1, said component receiving input from
at least one pen device driver.
8. (Currently Amended) A system for transferring information between unmanaged
code and managed code on a computing device, the unmanaged code being code native to and
executed directly by a processor of the computing device, the managed code being executed in a
common language run-time environment of a framework operating on the computing device, the
common language run-time environment of the framework executing the managed code
independent of a type of the processor of the computing device, the system comprising:
a mutual exclusion shared memory on the computing device designated to be non-
simultaneously shared between unmanaged code and managed code;
a pen input component on the computing device that is written in unmanaged code and
that receives pen data, the pen data being generated by a digitizer of the computing device upon
movement of a stylus with respect to a surface of the digitizer, the pen data including at least one
location on the digitizer of the stylus, the pen input component transferring information relating
to said pen data to said shared memory and transferring a pointer that points to said information

in the shared memory to a stylus input subsystem of an application on the computing device, the stylus input subsystem being separate from the pen input component and being having managed code;

said stylus input subsystem ~~application~~ having managed code receiving said pointer and obtaining said information from said shared memory by way of the transferred pointer.

9. (Original) The system according to claim 8, said component exposing a COM interface.

10. (Original) The system according to claim 8, said application using a P-Invoke-style command.

11. (Original) The system according to claim 8, said component including a pen services component.

12. (Original) The system according to claim 8, further comprising:
at least one pen device driver sending information to said component.

13. (Original) The system according to claim 8, further comprising:
said application including a pen input managed client.

14. (Currently Amended) A computer-readable storage medium having a program stored thereon for transferring information related to ink between an unmanaged pen input component and a managed stylus input subsystem of an application ~~including managed code~~ on a computing device, the unmanaged pen input component being native to and executed directly by a processor of the computing device, the managed stylus input subsystem ~~[[code]]~~ of the application being separate from the unmanaged pen input component and being executed in a common language run-time environment of a framework operating on the computing device, the

common language run-time environment of the framework executing the managed code independent of a type of the processor of the computing device, said program comprising the steps of:

receiving pen data in the unmanaged pen input component on the computing device, the pen data being generated by a digitizer of the computing device upon movement of a stylus with respect to a surface of the digitizer, the pen data including at least one location on the digitizer of the stylus;

transferring information related to said pen data to a mutual exclusion shared memory on the computing device designated to be shared non-simultaneously between the unmanaged pen input component `[[code]]` and the managed stylus input subsystem `[[code]]`;

transferring, by the unmanaged pen input component, a pointer that points to said information in said shared memory to the managed stylus input subsystem application; and

retrieving, by the managed stylus input subsystem ~~code of the application~~, said information from said shared memory by way of the transferred pointer.

15. (Previously Presented) The computer-readable storage medium according to claim 14, said program further comprising the steps of:

transferring additional information from said at least in part managed application to said shared memory;

transferring a pointer that points to said additional information to said component;

retrieving said additional information from said shared memory.

16. (Previously Presented) The computer-readable storage medium according to claim 14, said program further comprising the step of:

using a P-invoke style interface crossing between unmanaged code and managed code.

17. (Previously Presented) The computer-readable storage medium according to claim 14, said program further comprising the step of:

exchanging information through a COM interface.

18. (Previously Presented) The computer-readable storage medium according to claim 14, said component being a pen services component.

19. (Previously Presented) The computer-readable storage medium according to claim 14, said application including a pen input managed client.

20. (Previously Presented) The computer-readable storage medium according to claim 14, said component receiving input from at least one pen device driver.